Amendments to the Specification

Please replace the paragraph at page 11, line 24 page 12, line 11 with the following amended paragraph:

If the statement is found loop-invariant, at step 704 it is checked whether the statement is a load instruction and there is uncertain dependence (a possibility that a store instruction to the memory address from which the memory content is to be loaded may be executed in the loop). If so, the processing moves to step 705. Step 705 checks if the loop has already been duplicated (or copied). When the duplication of the loop is not yet made, the processing proceeds to step 706 where it duplicates the loop. This generates a copied loop at a position following the original loop. For example, the intermediate codes of Fig. 6 will be as shown in Fig. 8 after the step 706 is performed. In Fig. 8, BB5 (804) and BB6 (805) form the duplicated loop. Further, before the original loop a code (S15 in 807) for clearing the counter to zero is inserted.

Please replace the paragraph at page 12, line 20 - page 13, line 1 with the following amended paragraph:

Fig. 9 shows that a branch from the check instruction (S16 in 904) to the recovery codes in BB8 (906) is generated. At the start of the recovery codes in BB8, there is an instruction for incrementing the counter, S17: ctr = ctr + 1; ~~(S17 in 906)~~ and an instruction for branching to the duplicated loop when the counter exceeds a predetermined value, S18: if (ctr > T) (~~S18 in~~Step 906). When the counter does not exceed the predetermined value, a is reloaded by instruction S19: t1 = load.a(&a) at BB9 (~~S19 in~~Step 907) and the control returns to the instruction (S3 in 905) following the check instruction. At BB3, instructions S3: t2 = load(&b), S4: t3 = t1 + t2, S5: *p= t3, S6: t4 = load (p→next), and S7: p = t4 are executed (Step 905). At BB1, after Entry (901) proceed to BB7 where instructions S15: ctr = 0, and S2: t1 = load.a(&a) are executed (Step 902). Then at BB2, instruction S1: if (cond) is processed (Step 903) and at BB7, instruction S16: chk.a(t1) is executed (Step 904). If no branch at BB7, the proceed to BB3. At BB5 and instruction is executed, S8: if (cond) (Step 908) if not true, proceed to BB4 Exit (Step 910), else proceed to BB6 where instructions S9: t1 = load(&a), S10: t2 = load(&b), S11: t3 = t1 + t2, S12: *p = t3, S13: t4 = load (p→next), and S14: p=t4 are executed (Step 909), then loop back to BB5 (Step 908). Also, if at BB8 counter exceeds value, branch to BB6 (Step 909);

Please add the following new paragraphs at page 13, after line 22:

Fig. 8 shows a diagram of an example of intermediate codes immediately after the duplication of loop. After entry (801), a code (S15 in 807) for clearing the counter to zero is inserted. At BB2, includes instruction checking if cond, S1: if (cond) (Step 802), and if so proceed to BB3, if not proceed to BB4 Exit (806). At BB3, instructions S2: t1 = load(&a), S3: t2 = load(&b), S4: t3 = t1 + t2; S5: *p = t3, S6: t4 = load(p→next), and S7: p=t4 are performed (Step 803). At BB5 includes instruction checking if cond, S8: if (cond) (Step 804), and if so proceed to BB6, if not proceed to BB4 Exit (806). Further, at BB6, instructions S9: t1 = load(&a), S10: t2 =

load(&b), S11: t3 = t1 + t2, S12: *p = t3; S13: t4 = load (p→next), and S14: p = t4 are performed (Step 805).

Fig. 10 shows a diagram of an example of intermediate codes after the loop invariant code motion. After BB1, Entry (Step 1001), at BB7 instructions S15: ctr = 0, S2: t1 = load.a(&a), S3: t2 = load.a(&b), and S4: t3 = t1 + t2 are performed (Step 1002). At BB2, a condition is checked with instruction S1: if (cond) (Step 1003), and if not, proceed to BB6 where instructions S9: t1 = load(&a), S10: t2 = load(&b), S11: t3 = t1 + t2, S12: *p = t3, S13: t4 = load (p→next), and S14: p = t4 are processed (1012). If so, proceed to BB7, and a check S16: chk.a(t1) is processed (Step 1004). If no check, proceed to BB8 where S17: ctr = ctr + 1, and S18: if (ctr > T) is processed (Step 1007). If (ctr > T) then proceed to BB6, else to BB9 where instructions S19: t1 = load.a(&a), and S24: t3 = t1 + t2 are processed (Step 1008). If a check, proceed to BB 10 where another check instruction, S20: chk.a(t2) is processed (Step 1005). If no check at BB10, proceed to BB11 where instructions S21: ctr = ctr + 1, and S22: if (ctr > T) are processed (Step 1009) are processed. If (ctr > T) then proceed to BB6, else to BB12 where instructions S23: t2 = load.a(&b), and S25: t3 = t1 + t2 are processed (Step 1010). If a check at BB10, proceed to BB3 where instructions S5: *p = t3, S6: t4 = load(p→next), and S7: p = t4 are processed (Step 1006), then loop back to BB2. At BB5, a condition is checked processing instruction S8: if (cond) (Step 1011). If condition, then proceed to BB6 (Step 1020), else to BB4 (Step 1013).